```
; Program for picaxe micro AXE015X used in Dave-Rite mark 1 rebreather
; Hardware version 3.xx: software version x.51
; Picaxe programming editor v3.5.1
; Debug line 87


START:                           ;0
Pause 100                        ;1   LCD power up
EEPROM 0,("Dave-Rite v 3.51")    ;2   store the text in the EEPROM
goto INIT                        ;3   440

MAIN:                            ;4   print to LCD
for b3 = 0 to 15                 ;5   setup for...next loop ( 0 to 15)
read b3, b1                      ;6   read letter from EEPROM
gosub wrchr                      ;7   (sub 0) send character to LCD
next b3                          ;8   next loop
let b10 = 0                      ;9   Set menu steering variable to 0
pause 5000                       ;11
let b1 = 1                       ;12   Clear LCD
gosub wrins                      ;13  (gosub 1) 387
goto RECELL                      ;14

MEN:                             ;15  Menu
let b1 = 1                       ;16  Clear LCD
gosub wrins                      ;17  (gosub 2) 387
pause 1000                       ;18
let b1 = 192                     ;19  letter position on LCD
gosub wrins                      ;20  (gosub 3) 387
if b10 = 0 then L                ;21  Steering variable 0 = manual
if b10 = 2 then H                ;22  Steering variable 2 = low setpint
if b10 = 3 then M                ;23  Steering variable 3 = high
if b10 = 4 then L                ;24  Steering variable 4 = alarm mode
if b10 = 5 then H                ;25  Steering variable 5 = alarm mode
if b10 = 6 then L                ;26  Steering variable 6 = alarm low
if b10 = 7 then H                ;27  Steering variable 7 = alarm high
goto recell                      ;28

L:                               ;29  low setpoint
let b13 = 20                     ;30  Low setpoint loop time seconds*2
let b1 = 76                      ;31  low setpoint letter L
let b10 = 2                      ;32  Steering variable 2 = low
goto WRCHRR                      ;33  375
goto recell                      ;34

M:                               ;33  manual
let b13 = 120                    ;36  Loop count to 60 seconds about
let b1 = 77                      ;37  Manual letter M
let b10 = 0                      ;38  Steering variable 0 = manual
goto WRCHRR                      ;39  375
goto recell                      ;40

H:                               ;41  High setpoint
let b13 = 20                     ;42  High setpoint loop time seconds*2
let b1 = 72                      ;43  High setpoint letter H
let b10 = 3                      ;44  Steering variable 3 = high
goto WRCHRR                      ;45  375
goto recell                      ;46

RECELL:                          ;47  Read cell 0
for b12 = 1 to b13               ;48  Loop count b13 = loop time
```

```
readadc 0,b6                     ;49  Read cell 0 to b6
pause 25                         ;50
let b9 = b6                      ;51  Read cell to variable b9
let b1 = 128                     ;52  Cell 0 position on LCD
goto CELWRIT                     ;53  73

RECELLA:                         ;54  Read cell 1
let b0 = 1                       ;55  Read cell steering variable 1 = 1
readadc 1,b7                     ;56  Read cell 1 to b7
pause 25                         ;57
let b9 = b7                      ;58  Read cell to variable b9
let b1 = 134                     ;59  Cell 1 position on LCD
goto CELWRIT                     ;60  73

RECELLB:                         ;61  Read cell 2
let b0 = 2                       ;62  Read cell steering variable 2
readadc 2,b8                     ;63  Read cell 2 to b8
pause 25                         ;64
let b9 = b8                      ;65  Read cell to variable b9
let b1 = 140                     ;66  Cell 2 position on LCD
goto CELWRIT                     ;67  73

RECELLC:                         ;68  loop count displayed on LCD
let b0 = 3                       ;69  Read cell steering variable 3
let b9 = b12 / 2                 ;70  Save loop count to b9
let b1 = 205                     ;71  Loop count position on LCD
goto CELWRIT                     ;72  73

CELWRIT:                         ;73  Write cell value to LCD
gosub wrins                      ;74  (gosub 4) 387
pause 25                         ;75
gosub r                          ;76  (gosub 5) 219
pause 25                         ;77
if pin7 = 1 then MEN             ;78  Menu input switch
if b0 = 0 then RECELLA           ;79  Read cell steering variable 54
if b0 = 1 then RECELLB           ;80  Read cell steering variable 61
if b0 = 2 then RECELLC           ;81  Write loop count to LCD 68
if b0 = 3 then RECELLD           ;82  Alarm steering line 85
goto recell                      ;83

RECELLD:                         ;85  Menu steering
let b0 = 0                       ;86  steering variable 0 = cell 0
;debug b1                         ;87  temp debug read out
if b10 = 4 then AD               ;88  Steering variable alarm low set
if b10 = 5 then AD               ;89  Steering variable alarm mode high
if b10 = 6 then FLASH            ;90  Low ppo2 alarm low setpoint 411
if b10 = 7 then FLASH            ;91  Low pp02 alarm high setpoint 411
goto CON                         ;92  93

CON:                             ;93  Loop count
if b12 = b13 then COMP           ;94  97
next b12                         ;95  Next loop
goto RECELL                      ;96

COMP:                            ;97  Compare cell steering
if b10 = 0 then RECELL           ;98  Manual 47
let b6 = b6 + 20                 ; a  .2 offset
let b7 = b7 + 20                 ; a  .2 offset
let b8 = b8 + 20                 ; a  .2 offset
goto COMPP
```

```
COMPP:                          ;100 Compare cell for out of range
let b3 = b6 + 20                ;101 cell 0 high offset value
let b6 = b6 - 20                ;102 cell 0 low offset value
if b7 > b3 then ACEL            ;203 cell 1 compare to cell 0 high
if b7 < b6 then BCEL            ;104 cell 1 compare to cell 0 low
let b4 = b8 + 20                ;105 cell 2 high offset value
let b5 = b8 - 20                ;106 cell 2 low offset value
if b7 > b4 then CCEL            ;107 cell 1 compare to cell 2 high
if b7 < b5 then DCEL            ;108 cell 1 compare to cell 2 low
if b8 > b3 then ECEL            ;109 cell 2 compare to cell 0 high
if b8 < b6 then CCEL            ;110 cell 2 compare to cell 0,
let b6 = b6 + 20                ;111 Return b6 to cell reading
let b9 = b6 + b7 + b8 / 3       ;112 Average calculation for 3 cells
goto AVG                        ;113 114


AVG:                            ;114 Write average value to LCD
let b9 = b9 - 20                ; a  .2 offset
let b1 = 198                    ;115 Average position on LCD
gosub wrins                     ;116 (gosub 6)
gosub r                         ;117 (gosub 7) Write average value
if b10 = 6 then AVGL            ;118 Steering for low setpoint 249
if b10 = 2 then AVGL            ;119 Steering variable low setpoint
if b10 = 7 then AVGH            ;120 Steering variable high setpoint
if b10 = 3 then AVGH            ;121 Steering variable high setpoint
goto recell                     ;122

ACEL:                           ;123 cell 0 low
let b4 = b7 + 20                ;125 cell 1 high offset value
let b5 = b7 - 20                ;126 cell 1 low offset value
if b8 > b4 then AA              ;127 cell 2 compare to cell 1 high
if b8 < b5 then AA              ;128 cell 2 compare to cell 1 low
let b9 = b7 + b8 / 2            ;129 average cell 1 + 2
if b9 > b3 then CA              ;130 average cell 1 + 2 to cell 0 high
if b9 < b6 then AA              ;131 average cell 1 + 2 to cell 0 low
let b6 = b6 + 20                ;132 Return b6 to cell reading
let b9 = b6 + b7 + b8 / 3       ;133 Average calculation for 3 cells
goto AVG                        ;134 114

BCEL:                           ;135 cell 1 low
let b4 = b6 + 20                ;137 Restore cell reading store to
if b8 > b3 then AA              ;138 cell 2 compare to cell 0 high
if b8 < b6 then AA              ;139 cell 2 compare to cell 0 low
let b9 = b4 + b8 / 2            ;140 average cell 1 + 2
if b9 > b7 then CB              ;141 average cell 0 + 2 to cell 1 high
if b9 < b6 then AA              ;142 average cell 0 + 2 to cell 1 low
let b9 = b4 + b7 + b8 / 3       ;143 Average calculation for 3 cells
goto AVG                        ;144 114

CCEL:                           ;145 cell 2 low
let b6 = b6 + 20                ;147 Return b6 to cell reading
let b9 = b6 + b7 / 2            ;148 average cell 0 + 1
if b9 > b4 then CC              ;149 average cell 0 + 1 to cell 2 high
if b9 < b5 then AA              ;150 average cell 0 + 1 to cell 2 low
let b9 = b6 + b7 + b8 / 3       ;151 Average calculation for 3 cells
goto AVG                        ;152 114

DCEL:                           ;153 cell 1 low
if b8 > b3 then AA              ;155 cell 2 compare to cell 0 high
if b8 < b6 then AA               ;156 cell 2 compare to cell 0 low
let b6 = b6 + 20                 ;157 Return b6 to cell reading
let b9 = b6 + b8 / 2             ;158 average cell 1 + cell 2
let b4 = b7 + 20                 ;159 cell 1 high offset value
let b5 = b7 - 20                 ;160 cell 1 low offset value
if b9 > b4 then CB               ;161 average cell 0 + 2 to cell 1 high
if b9 < b6 then AA               ;162 average cell 0 + 2 to cell 1 low
let b9 = b6 + b7 + b8 / 3        ;163 Average calculation for 3 cells
goto AVG                         ;164 114

ECEL:                            ;165 253 cell 0 low
let b9 = b7 + b8 / 2             ;167 average cell 1 + cell 2
if b9 > b3 then CA               ;168 average cell 1 + 2 to cell 0 high
if b9 < b6 then AA               ;169 average cell 1 + 2 to cell 0 low
let b6 = b6 + 20                 ;170 Return b6 to cell reading
let b9 = b6 + b7 + b8 / 3        ;171 Average calculation for 3 cells
goto AVG                         ;172 114

CA:                              ;178
let b1 = 131                     ;179 # position on LCD
let pins = b1 & 240              ;180 Mask the high nibble of b1 b2
low 2                            ;181 Set RS low
pulsout 3,1                      ;182 Pulse the enable pin to send
pause 2                          ;183
let b2 = b1 * 16                 ;184 Put low nibble of b1 into b2.
let pins = b2 & 240              ;185 Mask the high nibble of b2
low 2                            ;186 Set RS low
pulsout 3,1                      ;187 Pulse enable pin to send data.
pause 2                          ;188
let b1 = 35                      ;189 ascii #
gosub wrchr                      ;190 (gosub 8) # to cell 0 363
goto AVG                         ;191 114

CB:                              ;192
let b1 = 137                     ;193 # position on LCD
let pins = b1 & 240              ;194 Mask the high nibble of b1  b2.
low 2                            ;195 Set RS low
pulsout 3,1                      ;196 Pulse the enable pin to send
pause 2                          ;197
let b2 = b1 * 16                 ;198 Put low nibble of b1 into b2.
let pins = b2 & 240              ;199 Mask the high nibble of b2
low 2                            ;200 Set RS low
pulsout 3,1                      ;201 Pulse enable pin to send data.
pause 2                          ;202
let b1 = 35                      ;203 ascii #
gosub wrchr                      ;203 (gosub 9) # to cell 1 363
goto AVG                         ;204 114

CC:                              ;205
let b1 = 143                     ;206 # position on LCD
let pins = b1 & 240              ;207 Mask the high nibble of b1  b2.
low 2                            ;208 Set RS low
pulsout 3,1                      ;209 Pulse the enable pin to send
pause 2                          ;210
let b2 = b1 * 16                 ;211 Put low nibble of b1 into b2.
let pins = b2 & 240              ;212 Mask the high nibble of b2
low 2                            ;213 Set RS low
pulsout 3,1                      ;214 Pulse enable pin to send data.
pause 2                          ;215
let b1 = 35                      ;216 ascii #
```

```
gosub wrchr                      ;217 (gosub 10) # to cell 2 363
goto AVG                         ;218 114


r:                               ;219 Write value to LCD subroutine
if pin7 = 1 then MEN             ;220 Menu input switch 15
if b9 < 100 then rr              ;221 Value less then 100 226
let b3 = b9 / 100                ;222 100th value
lookup b3,("012"),b3             ;223 Change binary value to ascii code
goto rrr                         ;224 233
return                           ;225

rr:                              ;226 Small value routine
let b3 = 48                      ;227 Set 100th value to ascii code
let b4 = b9 / 10                 ;228 10th value
lookup b4,("0123456789"),b4      ;229 Change 10th value to ascii code
let b5 = b9 // 10                ;230 1st value
lookup b5,("0123456789"),b5      ;231 Change 1st value to ascii code
goto dis                         ;232 242

rrr:                             ;233 Large value routine
let b4 = b9 - 100                ;234 Subtract 100th value
let b4 = b4 / 10                 ;235 10th value
lookup b4,("0123456789"),b4      ;236 Change 10th value to ascii code
let b5 = b9 - 100                ;237 Subtract 100th value
let b5 = b5 // 10                ;238 1st value
lookup b5,("0123456789"),b5      ;239 Change 1st value to ascii code
goto dis                         ;240 242

dis:                             ;241 Write ascii cell value to LCD
let b1 = b3                      ;242
gosub wrchr                      ;243(gosub 11) Write character 100th
let b1 = b4                      ;244
gosub wrchr                      ;245(gosub 12) Write character 10th
let b1 = b5                      ;246
gosub wrchr                      ;247(gosub 13) Write character 1st 363
return                           ;248

AVGL:                            ;249 Low setpoint
if b9 > 140 then AA              ;250 compare average to high setpoint,
if b9 < 50 then INJL             ;251 compare average to 02 setpint
let b10 = 2                      ;252 Steering variable low setpoint
goto RECELL                      ;253

INJL:                            ;254 Low setpoint table
if b9 <= 30 then INJLC           ;255 270
if b9 <= 40 then INJLB           ;256 263
let b13 = 30                     ;257 Loop time variable
high 1                           ;258 Sol output
pause 400                        ;259 Injection time
low 1                            ;260 Sol output reset
let b10 = 2                      ;261 Menu steering variable
goto recell                      ;262 47

INJLB:                           ;263 Injection time B
let b13 = 30                     ;264 Loop time variable
high 1                           ;265 Sol output
pause 800                        ;266 Injection time
low 1                            ;267 Sol output reset
let b10 = 2                      ;268 Menu steering variable
```

```
goto recell                      ;269 47

INJLC:                           ;270 Injection time C
let b13 = 18                     ;271 Loop time variable
high 1                           ;272 Sol output
pause 1200                       ;273 Injection time
low 1                            ;274 Sol output reset
let b10 = 6                      ;275 Menu steering variable
goto SIG                         ;276 Low pp02 alarm 350
goto recell                      ;277


AVGH:                            ;278 High setpoint
if b9 > 150 then AB              ;279 compare average to high setpoint,
if b9 < 130 then INJH            ;280 compare average to 02 setpint
let b10 = 3                      ;281
goto RECELL                      ;282 47

INJH:                            ;283 High setpoint table
if b9 <= 70 then INJHF           ;284 342
if b9 <= 95 then INJHE           ;285 324
if b9 <= 105 then INJHD          ;286 317
if b9 <= 110 then INJHC          ;287 310
if b9 <= 115 then INJHB          ;288 303
if b9 <= 120 then INJHA          ;289 296
let b13 = 30                     ;290 Loop time variable
high 1                           ;291 Sol output
pause 400                        ;292 Injection time
low 1                            ;293 Sol output reset
let b10 = 3                      ;294 Menu steering variable
goto recell                      ;295 47

INJHA:                           ;296 Injection time A
let b13 = 30                     ;297 Loop time variable
high 1                           ;298 Sol output
pause 700                        ;299 Injection time
low 1                            ;300 Sol output reset
let b10 = 3                      ;301 Menu steering variable
goto recell                      ;302 47

INJHB:                           ;303 Injection time B
let b13 = 30                     ;304 Loop time variable
high 1                           ;305 Sol output
pause 1000                       ;306 Injection time
low 1                            ;307 Sol output reset
let b10 = 3                      ;308 Menu steering variable
goto recell                      ;309 47

INJHC:                           ;310 Injection time C
let b13 = 30                     ;311 Loop time variable
high 1                           ;312 Sol output
pause 1200                       ;313 Injection time
low 1                            ;314 Sol output reset
let b10 = 3                      ;315 Menu steering variable
goto recell                      ;316 47

INJHD:                           ;317 Injection time D
let b13 = 20                     ;318 Loop time variable
high 1                           ;319 Sol output
pause 1400                       ;320 Injection time
```

```
low 1                            ;321 Sol output reset
let b10 = 3                      ;322 Menu steering variable
goto recell                      ;323 47

INJHE:                           ;324 Injection time E
let b13 = 18                     ;325 Loop time variable
high 1                           ;326 Sol output
pause 1600                       ;327 Injection time
low 1                            ;328 Sol output reset
let b10 = 7                      ;329 Menu steering variable
goto SIG                         ;340 Low pp02 alarm
goto RECELL                      ;341 47

INJHF:                           ;342 Injection time F
let b13 = 18                     ;343 Loop time variable
high 1                           ;344 Sol output
pause 1800                       ;345 Injection time
low 1                            ;346 Sol output reset
let b10 = 7                      ;347 Menu steering variable
goto SIG                         ;348 Low pp02 alarm
goto RECELL                      ;349 47

SIG:
let b1 = 193                     ;350 Letter position on LCD
let pins = b1 & 240              ;351 Mask the high nibble of b1 b2.
low 2                            ;352 Set RS low
pulsout 3,1                      ;353 Pulse the enable pin to send
pause 2                          ;354
let b2 = b1 * 16                 ;355 Put low nibble of b1 into b2.
let pins = b2 & 240              ;356 Mask the high nibble of b2
low 2                            ;357 Set RS low
pulsout 3,1                      ;358 Pulse enable pin to send data.
pause 2                          ;359
let b1 = 35                      ;360 ascii #
gosub wrchr                      ;361 (gosub 14) # to alarm position
goto RECELL                      ;362 47

wrchr:                           ;363 Write characters to LCD (subroutine)
let pins = b1 & 240              ;364 Mask the high nibble of b1 b2.
high 2                           ;365 Set RS high
pulsout 3,1                      ;366 Pulse the enable pin to send
pause 2                          ;367
let b2 = b1 * 16                 ;368 Put low nibble of b1 into b2.
let pins = b2 & 240              ;369 Mask the high nibble of b2
high 2                           ;370 Set RS high
pause 1                          ;371 Make sure RS is high
pulsout 3,1                      ;372 Pulse enable pin to send data.
pause 2                          ;373
return                           ;374

WRCHRR:                          ;375 Write instruction to LCD (menu
let pins = b1 & 240              ;376 Mask the high nibble of b1 b2.
high 2                           ;377 Set RS high
pulsout 3,1                      ;378 Pulse the enable pin to send
pause 1                          ;379
let b2 = b1 * 16                 ;380 Put low nibble of b1 into b2.
let pins = b2 & 240              ;381 Mask the high nibble of b2
high 2                           ;382 Set RS high
pause 1                          ;383 Wait 1 ms
pulsout 3,1                      ;384 Pulse enable pin to send data.
```

```
pause 1500                      ;385 Wait 2 seconds
goto RECELL                     ;386 47

wrins:                          ;387 Write instruction to LCD
let pins = b1 & 240             ;388 Mask the high nibble of b1 b2.
low 2                           ;389 Set RS low
pulsout 3,1                     ;390 Pulse the enable pin to send
pause 2                         ;391
let b2 = b1 * 16                ;392 Put low nibble of b1 into b2.
let pins = b2 & 240             ;393 Mask the high nibble of b2
low 2                           ;394 Set RS low
pause 2                         ;395
pulsout 3,1                     ;396 Pulse enable pin to send data.
pause 2                         ;397
return                          ;398

WRINSS:                         ;399 Write instruction to LCD routine
let pins = b1 & 240             ;400 Mask the high nibble of b1  b2.
low 2                           ;401 Set RS low
pulsout 3,1                     ;402 Pulse the enable pin to send
pause 1                         ;403
let b2 = b1 * 16                ;404 Put low nibble of b1 into b2.
let pins = b2 & 240             ;405 Mask the high nibble of b2
low 2                           ;406 Set RS low
pause 1                         ;407
pulsout 3,1                     ;408 Pulse enable pin to send data.
pause 1                         ;409
goto AC                         ;410 426

FLASH:                          ;411 Low pp02 alarm
high 0                          ;412 LED output
pause 100                       ;413 LED on time
low 0                           ;414 LED off
goto CON                        ;415 93

AA:                             ;416 alarm
let b10 = 4                     ;417 Steering variable 4 = alarm mode
let b1 = 192                    ;418 Letter position
goto WRINSS                     ;419 Write instruction routine AC 399
goto recell                     ;420

AB:                             ;421 alarm
let b10 = 5                     ;422 Steering variable 5 = alarm mode
let b1 = 192                    ;423 Letter position
goto WRINSS                     ;424 Write instruction routine AC 399
goto recell                     ;425

AC:                             ;426
let b1 = 67                     ;427 Letter C
goto WRCHRR                     ;428 375
goto recell                     ;429

AD:                             ;430 Alarm output
if pin7 = 1 then MEN            ;431 Switch input go to menu 15
high 0                          ;432 Alarm LED
pause 150                       ;433
low 0                           ;434 Alarm LED
pause 150                       ;435
high 0                          ;436 Alarm LED
pause 150                       ;437
```

```
low 0                             ;438 Alarm LED
goto RECELL                       ;439 47

INIT:                             ;440 Initialize LCD
let pins = 0                      ;441 Clear all output lines
pause 200                         ;442 Wait 200 ms for LCD to reset.
let pins = 48                     ;443 Set to 8-bit operation.
pulsout 3,1                       ;444 Send data by pulsing .enable.
pause 10                          ;445 Wait 10 ms
pulsout 3,1                       ;446 Send data again
pause 5                           ;447 Wait 5 ms
pulsout 3,1                       ;448 Send data again
let pins = 32                     ;449 Set to 4-bit operation.
pulsout 3,1                       ;450 Send data.
pause 5                           ;451 Wait 5 ms
pulsout 3,1                       ;452 Send data again.
let pins = 128                    ;453 Set to two line operation
pulsout 3,1                       ;454 Send data.
pause 5                           ;455
let b1 = 12                       ;456 Screen on, cursor on instruction
let pins = b1 & 240               ;457 Mask the high nibble of b1 b2.
low 2                             ;458 Set RS low
pulsout 3,1                       ;459 Pulse the enable pin to send
let b2 = b1 * 16                  ;460 Put low nibble of b1 into b2.
let pins = b2 & 240               ;461 Mask the high nibble of b2
low 2                             ;462 Set RS low
pause 1                           ;463 Wait 1 ms
pulsout 3,1                       ;464 Pulse enable pin to send data.
high 2                            ;465 Back to character mode
goto MAIN                         ;466 4
```

Code written by Dave the Homebuilder 2004
Download from: http://www.therebreathersite.nl

These data are only for study purposes.